WL-TM-96-4059

V-LAB: A Virtual Laboratory for Structural Integrity Assessment of Composite Components

Applied Research Associates

Corporate Headquarters
4300 San Mateo Blvd, NE, Suite A220
Albuquerque, NM 87110
505-881-8074

Southeast (Performing) Division
6406 Falls of Neuse Road
Raleigh SC
919-8786-0018

31 JANUARY 1996

FINAL REPORT FOR 4/26/95 -- 1/1/96

Prepared for:

Department of the Air Force
Wright Laboratory (AFMC)
Wright-Patterson AFB, Ohio

MATERIALS DIRECTORATE
WRIGHT LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7734

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

The technical report has been reviewed and is approved for publication.

Patricia L. Stumpff

PATRICIA L. STUMPFF
Struct & Elect FA Section
Materials Integrity Branch
Systems Support Division
Materials Directorate

Ronald H. Williams

RONALD H. WILLIAMS, Br Chf
Materials Integrity Branch
Systems Support Division
Materials Directorate

Gary K. Waggoner

GARY K. WAGGONER, Chief
Systems Support Division
Materials Directorate

If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify WL/MLSA, Bldg 652, 2179 Twelfth St, Ste 1, WPAFB OH 45433-7718 to help maintain a current mailing list.

Copies of this report should not be returned unless return is required by Security Consideration, contractual obligations, or notice on a specific document.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
31 January 1996

3. REPORT TYPE AND DATES COVERED
FINAL 26 Apr 95--31 Jan 96

4. TITLE AND SUBTITLE

V-LAB: A Virtual Laboratory for Structural Integrity Assessment of Composite Components

5. FUNDING NUMBERS

C F33615-95-C-5624
PE 65502
PR 3005
TA 05
WU F3

6. AUTHOR(S)

Eduardo Moas and Timothy R. Wyatt

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Applied Research Associates, Inc.
6404 Falls of Neuse Road, Suite 200
Raleigh NC 27615

8. PERFORMING ORGANIZATION REPORT NUMBER

5278

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Materials Directorate
Wright Laboratory
Air Force Materiel Command
Wright Patterson Air Force Base, Ohio 45433-7718

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

WL-TR-96-4059

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for Public Release; Distribution is Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

V-LAB is a Window's-based analysis tool for structural integrity assessment of composite components. The software allows failure investigators to work within "virtual laboratories" representing different structural levels within a composite component. Data can be exchanged seamlessly between the different labs, allowing results to be easily integrated. A prototype of the V-LAB software system was developed with labs for micromechanics analysis, material definition, and failure analysis of composite laminates and thin-walled beams.

V-LAB provides a general software framework where independent analysis modules can be coupled for structural analysis. A relational database was developed containing all required data for the various analyses. The independent modules can share information via the database, and new modules can be added to V-LAB. Access is provided to existing material databases and analysis functions, simplifying development of new modules.

This research demonstrated the potential to develop an easy-to-use, integrated analysis environment that meets the Air Force's need to perform efficient, independent failure analyses. Due to the flexible, extendible nature of the program, V-LAB can be easily adapted for use by civilian agencies and industries involved in design and failure analysis of composite components.

14. SUBJECT TERMS

Classical Lamination Theory, Composites, Micromechanics, Object-Oriented Programming, Phenomenological Failure Criterion, Relational Database, Structures, Thin-Walled Beams, Windows Aplication

15. NUMBER OF PAGES

36

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

V-LAB: A VIRTUAL LABORATORY FOR STRUCTURAL INTEGRITY ASSESSMENT OF COMPOSITE COMPONENTS

TABLE OF CONTENTS

1.0 Executive Summary

1.1 Background

The use of advanced composite materials for military airframe structures has been continually increasing due to weight savings and improved strength and corrosion resistance of composites compared to metallic components. However, composite structures can fail prematurely due to gross manufacturing defects, design errors, or severe in-service damage. Investigations of these failed components often involve stress analysis of the components to assess their structural integrity, both with and without damage.

In the past, the Air Force has relied on contractor support for these stress analyses. These are component specific analyses which often must be modified to account for deficiencies noted during an investigation. In order to conduct a complete and unbiased investigation, the Air Force requires a fast, general purpose, easy-to-use analysis tool that is capable of assessing the structural integrity of a wide range of composite components. Currently there exist no general-purpose commercial or public domain software packages for composite structural analysis that are accessible to failure investigators who are not experts in these types of analyses.

The overall goal of the Phase I research effort was to investigate the feasibility of developing a user-friendly, PC-based analysis tool for assessing the structural integrity of composite components. The analysis tool is a set of virtual laboratories (V-LAB) for conducting analyses at different structural "levels", as illustrated in Figure 1, including micromechanics, individual lamina or plies, laminated composites, and structural applications. The Windows-based software includes an intuitive user interface that allows you to work within each lab independently, or share data between labs.

1.2 Phase I Technical Objectives

The Phase I effort demonstrated the technical feasibility of the proposed virtual lab approach for performing failure investigations of composite components. The specific technical objectives for Phase I were as follows:

(1) Identify analysis methodologies to incorporate in V-LAB.

(2) Develop object-oriented software framework.

(3) Design intuitive graphical user interface.

(4) Implement and test V-LAB software system.

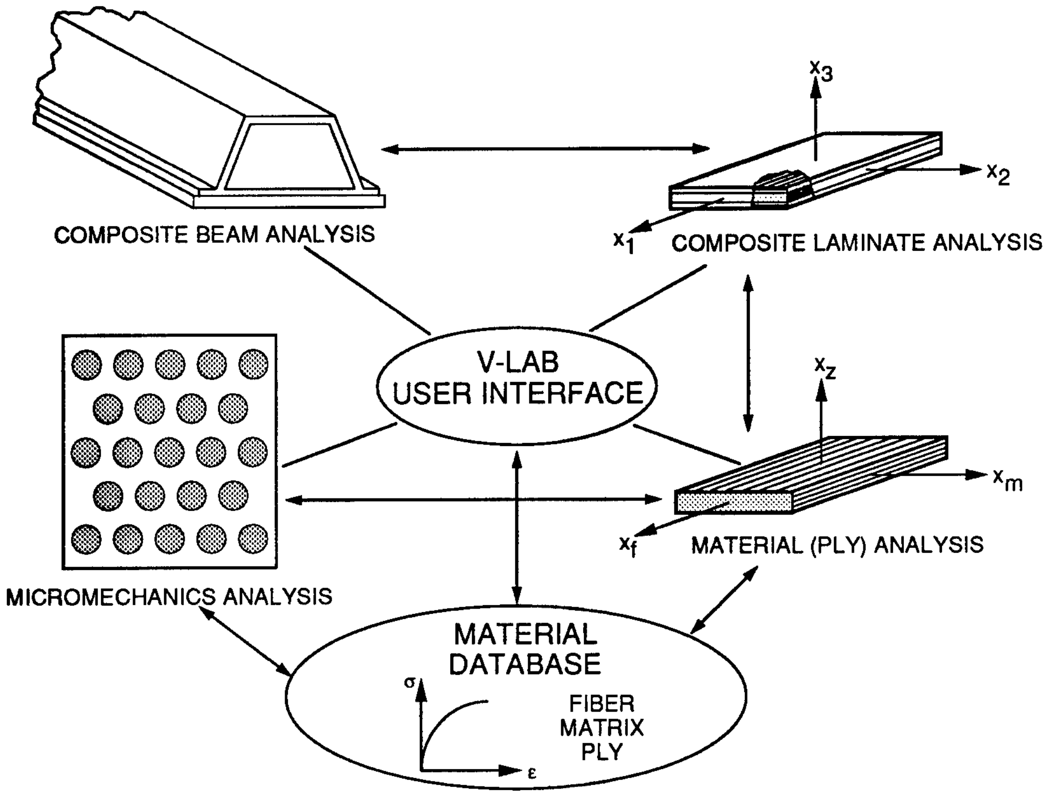


Figure 1—V-LAB Performs Integrated Analyses at Several Structural Levels

1.3 Phase I Accomplishments

The Phase I effort successfully demonstrated the technical feasibility of the V-LAB concept and met all the technical objectives set forth for Phase I. V-LAB is operational with four integrated analysis labs: a Micromechanics Lab, Materials Lab, Composites Lab, and Beam Analysis Lab. You can use different labs to conduct independent analyses, and exchange results seamlessly between the labs as illustrated in Figure 2. The Materials Lab analysis window on the left side of the figure is used for general constitutive and failure analysis at a material point (e.g. typically used to characterize the lamina, or ply, response). You can access material properties from the V-LAB database, define a new material via direct data input, or import material data from a previous V-LAB session. Another method to define the material properties is via a micromechanics analysis, which can be launched using the micromechanics button in the Materials Lab window. The micromechanics analysis window, shown on the right side of the figure, computes the ply elastic constants based on the constituent (fiber/matrix) properties using a numerical solution of the governing elasticity equations. Again you can obtain data directly from the material database, specify your own constituent properties, or import data defined previously. After the micromechanics computations are completed, the new ply constants are updated in the Materials Lab window, and in the materials database.

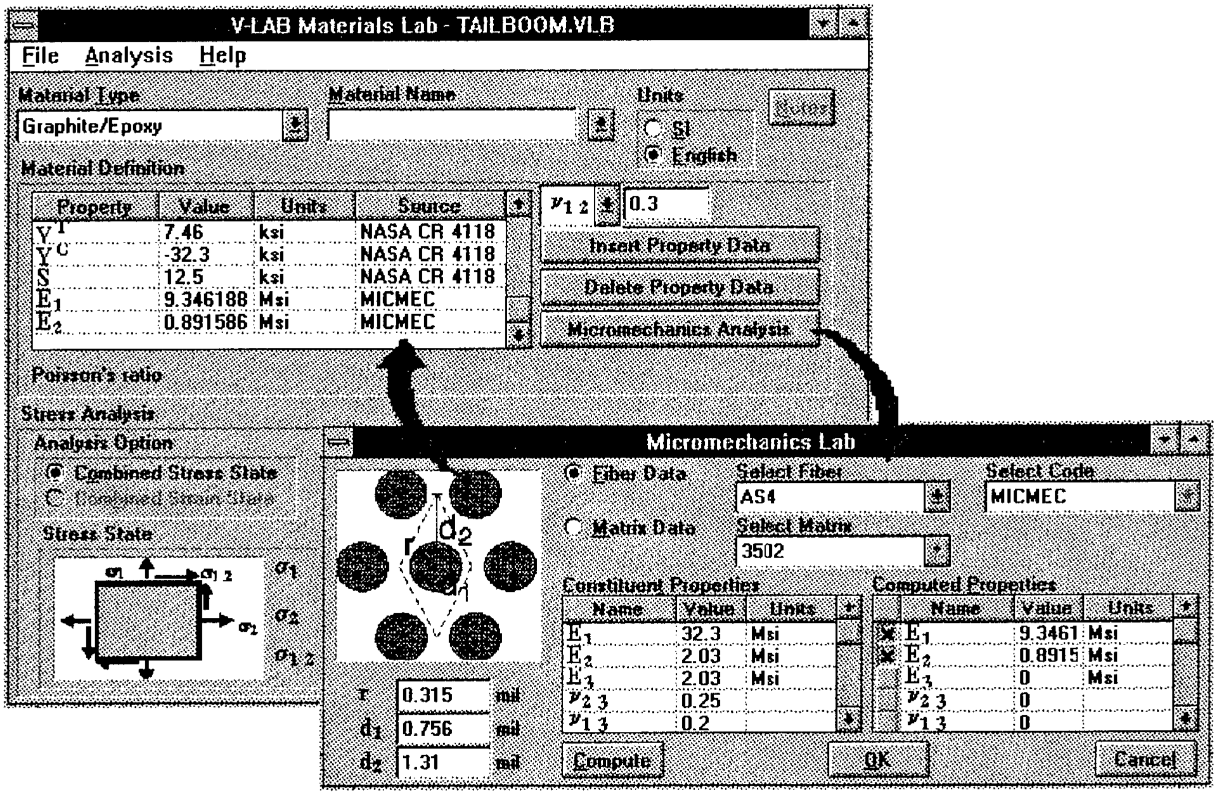


Figure 2—The Micromechanics Model in V-LAB is Seamlessly Integrated with Materials Analysis

Specific results from the Phase I research are listed below, grouped by the technical objectives of the Phase I project.

(1) Identified analysis methodologies. We identified suitable analysis methods to include in V-LAB to enable the structural integrity assessment of composite components. Table 1 lists a breakdown of these methods by analysis type, e.g., micromechanics, laminate analysis, and structural applications. We were assisted in this task by Northrop Grumman Corporation and Durability, Inc. Northrop Grumman provided valuable input on their analysis methods and capabilities for design and failure of composite components. These codes, representing industry standards for composites analysis, will be incorporated in V-LAB in Phase II. Durability provided a comprehensive literature review of micromechanics methods, which will be used to enhance the Micromechanics Lab in Phase II. V-LAB currently includes integrated analysis capabilities for micromechanics, material (ply), composite laminate, and beam sections as listed in Table 1. The functionality implemented in Phase I was sufficient to demonstrate how these integrated analysis modules could be used to aid in a failure investigation; V-LAB was demonstrated with an Air Force case study of a cylindrical, graphite/epoxy tail boom that failed due to a gross manufacturing defect (see Section 4.0 for details of analysis).

TABLE 1—CURRENT AND PROPOSED V-LAB COMPOSITES ANALYSIS CAPABILITIES

Analysis Description	Phase I	Phase II
Micromechanics Models		
diamond-packed array	✔	
rectangular-packed array		✔
compute ply properties	✔	
micromechanical stresses		✔
coatings		✔
failure analysis		✔
Material constitutive models		
linear elastic	✔	
nonlinear elastic		✔
thermal effects		✔
moisture effects		✔
Material failure models		
maximum strain	✔	
Tsai-Wu	✔	
higher-order tensor polynomial		✔
Laminated Composite Models		
ply-by-ply stresses (via classical lamination theory)	✔	
elasticity solution		✔
free-edge interlaminar stress analysis		✔
disbond/delamination analysis		✔
moisture diffusion		✔
Beam Structural Analysis		
section properties calculations	✔	
stress analysis	✔	
failure analysis	✔	
thin-walled beams (shear flow)		✔
Graphical Display		
through-thickness stress distribution	✔	
failure surfaces	✔	
carpet plots of composite laminate properties		✔
Structural Application		
beams	✔	
plate with cutout		✔
bonded joints		✔

(2) Developed software framework. A relational database schema was designed and implemented that allows data to be defined for individual labs, and shared seamlessly among other labs. The schema is based on an optimal normal form that ensures the data integrity while allowing the database to be easily extendible. New materials can be added to the database, either by modifying existing materials or defining new ones. In addition, a novel “session file” concept

was implemented allowing you to save data and results from a current V-LAB session for later use, and to import data from previous V-LAB sessions into a current session.

The database provides the foundation for the V-LAB software framework, as illustrated in Figure 3. You interact with the graphical user interface (GUI), which controls the underlying analysis modules directly. The GUI can update information in the database directly, via direct user input, or an analysis module can compute data that is updated in the database, e.g. the micromechanics module updating ply elastic constants in Figure 2. The various analysis modules can be run separately, each controlled by its own analysis window, yet they can communicate with one another via the database. For example, a beam analysis may have access to all laminates defined via the Composites Lab; the Composites Lab may use materials defined via the Materials Lab, etc.

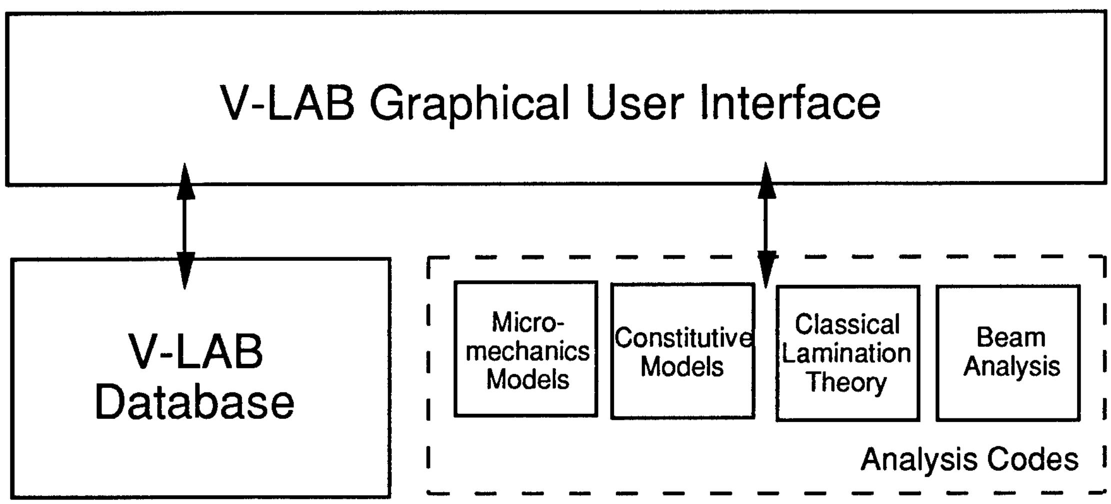
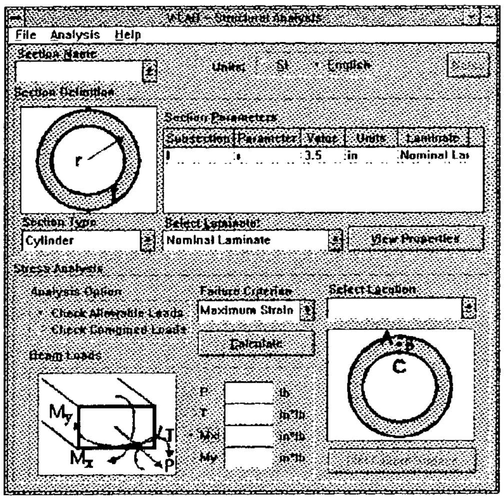


Figure 3—Relational Database Enables Communication Between Analysis Modules

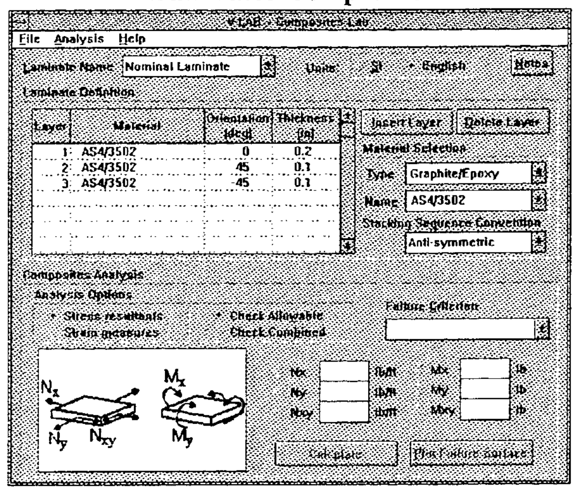
(3) Designed intuitive graphical user interface. An intuitive, Windows-based interface was designed that allows easy definition and analysis of the problem. Each analysis lab has a separate window, shown in Figure 4. The program documentation is "on-line" and can be activated by a context-sensitive Help feature available for each screen. The screens were developed to have a consistent "look and feel": the top part of each screen is used to define the problem parameters, and the bottom part of the screen controls the analysis. The results of analyses are generally displayed graphically instead of relying on tabulated results. For example, Figure 5 illustrates the through-thickness stress distribution and failure envelope calculated by the Composites Lab. The distribution of stresses through the laminate thickness is computed using classical lamination theory and displayed as three plots for σ_1, σ_2, σ_{12}. A slider allows you to specify a location in the composite laminate z-direction, and the stress values at the location is printed. Additionally, the

failure surface is calculated for the given stress state and the point on the failure surface is identified.

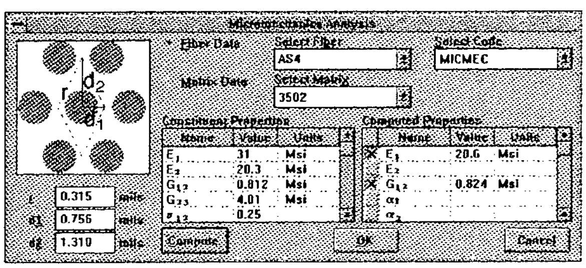
Beam Analysis



Laminated Composites



Micromechanics Analysis



Materials Analysis

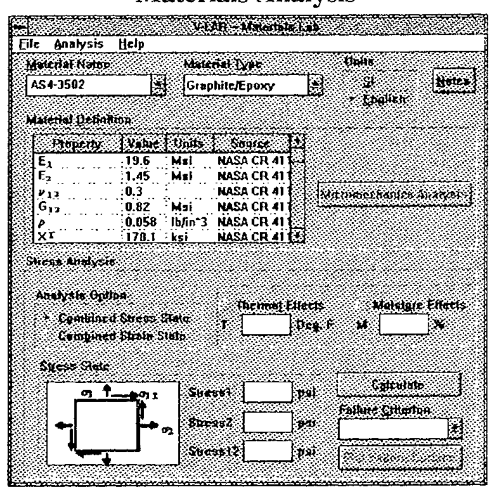


Figure 4—The V-LAB Analysis Environment Provides an Intuitive User Interface

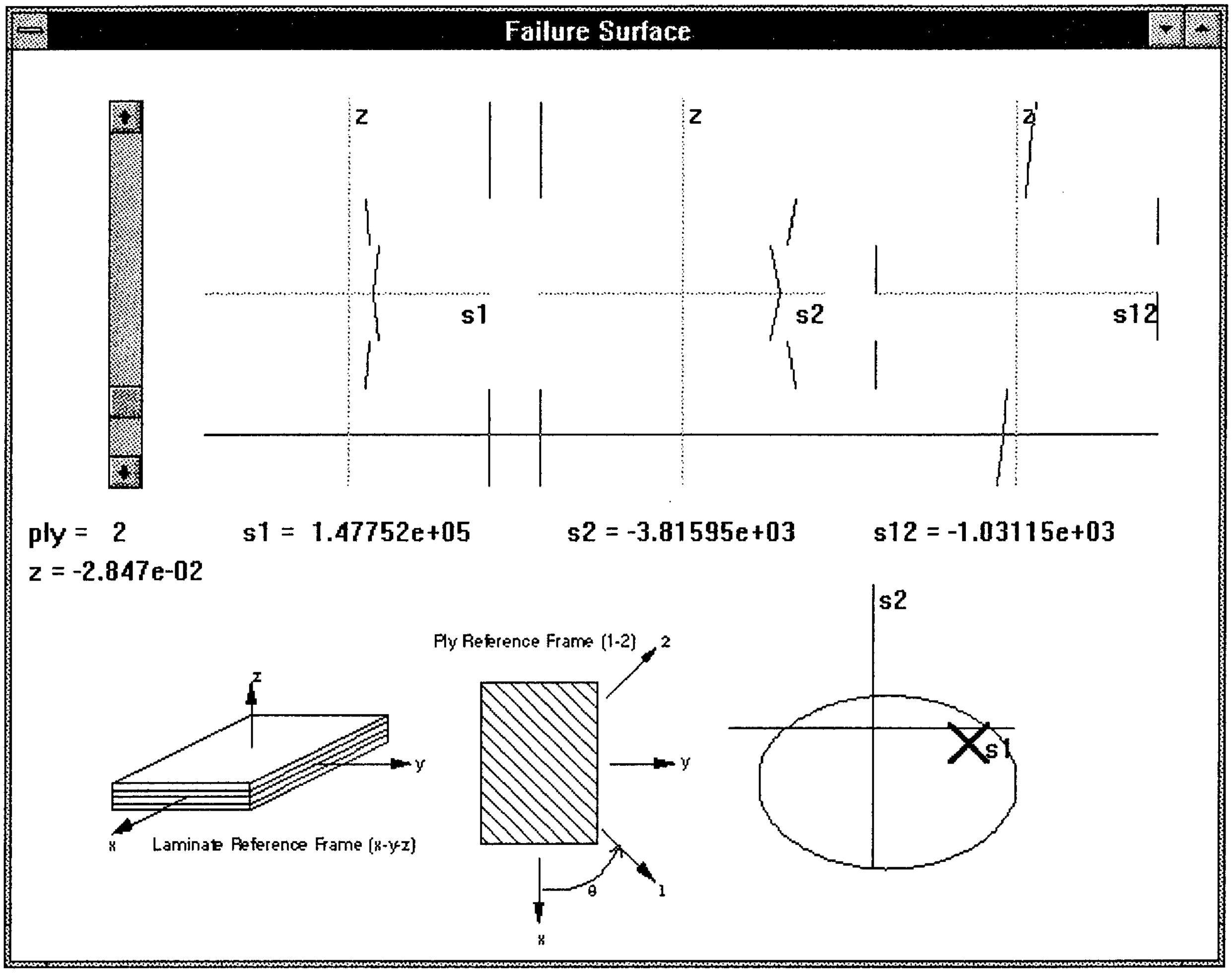


Figure 5—V-LAB Provides Graphical Display of Laminate Through-Thickness Stress Distribution and Failure Surface

(4) Implement and test V-LAB software. The V-LAB software system was successfully implemented in Phase I, including the database, session file features, analysis modules listed in Table 1, the four lab screens shown in Figure 4, the stress and failure analysis screen pictured in Figure 5, and the on-line documentation. All the individual analysis components were tested during implementation by comparing the results to existing codes. The integrated software system was demonstrated with an Air Force case study of a Gr/Ep cylindrical tail boom that failed due in part to gross manufacturing defects. Using the Composites Lab, we modeled both the nominal and "as-manufactured" laminate configurations, accessing material data from the database.[1] We modeled the tail boom section in the Structures Lab, and analyzed the allowable loads using both the nominal and "as-manufactured" laminate configurations. The analysis indicated a 7x reduction in load-carrying capability for the "as-manufactured" laminate,

[1] The integrated micromechanics capabilities were not used in the demonstration, but potentially could be useful for incorporating additional material information from typical fractographic investigations.

indicating a potential cause of failure in the tail boom. See Section 4.0 for details on this case study.

1.4 Payoff

V-LAB provides access to material databases and analysis codes in a seamless, integrated analysis environment. In Phase I we demonstrated V-LAB with an Air Force case study of a Gr/Ep cylindrical tail boom that failed due in part to gross manufacturing defects. Typically, such an analysis would be completed with the aid of material handbooks, a micromechanics code, a laminate analysis code, and closed form beam solutions. The Windows-based software allowed these analyses to be done with point-and-click ease while hiding all the data exchange behind the scenes.

This research demonstrated the potential to develop a simple, integrated analysis environment to perform structural integrity assessment of composite components that will meet the Air Force's need in performing efficient, independent failure analyses. Due to the flexible, extendible nature of the program, V-LAB can be easily adapted for use by the FAA for failure analysis of components from commercial aircraft. The Air Force has supported several NTSB failure investigations for commercial aircraft, which should accelerate future transfer of this Air Force technology to the commercial sector.

Currently no commercial programs exist with the full range of capabilities demonstrated for V-LAB in a tightly integrated, flexible, easy to use Windows environment. Aerospace manufacturers rely on design manuals and specialized analysis codes for analysis of composite components. An integrated tool such as V-LAB could be used early in the design stage of a component to quickly analyze prototypes, and could be extended to incorporate a company's particular analysis or database. Besides the aerospace industry, V-LAB has potential use in other markets that are heavily exploiting the use of composites, like the sports-equipment, automotive and shipbuilding industries.

V-LAB has strong commercial viability, as evidenced by our securing a software distribution agreement with Technomic Publishing Company, a major distributor of composites software with a customer base of over 7000 composites users. In Phase III, ARA has agreed to jointly produce and market the V-LAB software system with Technomic. Michael Margotta, the president and CEO of Technomic, has said that V-LAB surpasses currently available composites software, and that V-LAB would replace several of his currently available composites analysis codes.

2.0 Analysis Capabilities

The V-LAB code is designed to allow the investigator to perform failure analysis of laminated composite components at several structural "levels": material (ply), laminate, and composite beam. Each of these analyses is controlled by a separate analysis screen in the V-LAB Windows environment.

Material Analysis. The Materials Lab is used to define material properties, and includes a micromechanics function to compute the properties via a series-type elasticity solution. A phenomenological failure criterion is used to evaluate the failure condition for a given stress state (i.e. failed/safe). The failure surface can be plotted using the materials analysis screen.

Laminate Analysis. The Composites Lab is used to define laminates composed of plies defined in the Materials Analysis Lab. The Composites Lab performs calculations based on classical lamination theory to compute effective laminate properties, and to recover pointwise stresses in individual plies. The Composites Lab includes a function to compute allowable loads on the laminate, and to plot the corresponding through-thickness (ply-by-ply) stress distribution and failure surfaces.

Composite Beam Analysis. The Beam Lab is used to define and analyze beam sections composed of laminates defined in the Composites Lab. Several beam section types are supported, including rectangular, box, I, C, and cylindrical shell. Beam section properties (modulus-weighted) can be calculated, as well as allowable loads. The through-thickness stress distribution and failure surfaces can be plotted in the laminate where the critical failure occurs.

2.1 Materials Lab

The Materials Lab is used to define the mechanical properties and failure data at the ply level, which represents the basic building block within the current V-LAB analysis environment. The material properties may be specified by one of several methods, including (1) accessing or modifying material data in the V-LAB database; (2) direct user input; (3) importing data from a previous V-LAB session; and (4) via a micromechanics analysis. The Tsai-Wu failure criterion is used to construct a fail/safe envelope for the material in stress space; stress states that fall within the envelope are "safe", while those that are outside the envelope are "failed". Some details on the micromechanics analysis and Tsai-Wu failure criterion are described next.

2.1.1 Micromechanics Analysis

The micromechanics analysis is based on the MICMEC code developed by Averill and Carman [1991], and assumes the fibers are packed in a diamond array. Due to symmetry a single fiber

can be isolated in a representative volume element (RVE) as shown in Fig. 6. MICMEC can model an arbitrary number of phases within an RVE, e.g. the fiber, fiber coatings (interphase), matrix, etc. In the V-LAB implementation of MICMEC, the number of phases is limited to two: the fiber and matrix materials. The MICMEC code performs a series-type elasticity solution for diamond-packed fiber reinforced composites. This solution accounts for all six components of mechanical loading, phase-dependent hygro-thermal loading, cylindrically orthotropic, and temperature- and moisture-dependent constitutive properties.

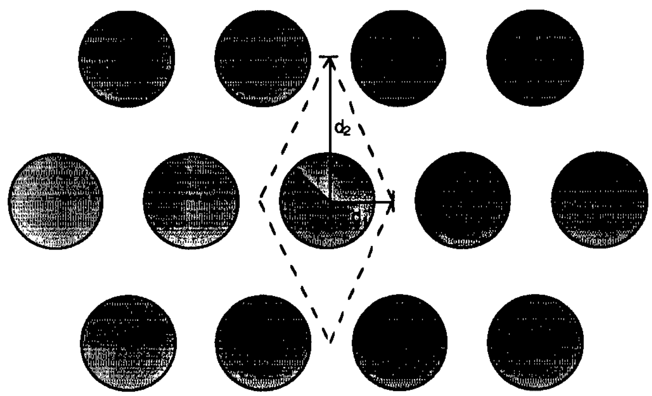


Figure 6—Representative Volume Element in Diamond-Packed Fiber Array

2.1.2 Tsai-Wu Failure Criterion

Analysis of the failure phenomenon in composites can be considered from a mechanistic or a phenomenological point of view. Mechanistic theories model the mechanics of the composite's constituent materials to obtain the physical properties and response of the composite, e.g. the micromechanics analysis described above. Phenomenological failure theories relate mathematical models to the failure phenomenon and generally depend on empirical curve-fitting to determine the model parameters (see [Nahas 1986] for a comprehensive review).

Phenomenological failure criterion were chosen for V-LAB due to their computational efficiency. The widely use Tsai-Wu criterion was implemented in the current version of V-LAB. Tsai and Wu [1971] proposed a failure criterion in the form of a quadratic tensor polynomial:

$$f\left(\sigma_I, F_I, F_{IJ}\right) = F_I \sigma_I + F_{IJ}\,\sigma_I\,\sigma_J = 1 \qquad \mathrm{I,J} = 1,2,\ldots 6 \qquad (1)$$

where σ_I are components of the second order stress tensor in contracted notation, and F_I, F_{IJ} are components of the second and fourth order strength tensors. The function *f* is a scalar quantity and is invariant to coordinate transformations, i.e. the strength tensors F_I, F_{IJ} can be transformed to an arbitrary reference frame.

For the case of plane stress ($\sigma_3 = \sigma_4 = \sigma_5 = 0$), Eq. 1 reduces to:

$$
\begin{aligned}
f = F_1\sigma_1 + F_2\sigma_2 + F_6\sigma_6 + F_{11}\sigma_1^2 + 2F_{12}\sigma_1\sigma_2 + 2F_{16}\sigma_1\sigma_6 \\
+ F_{22}\sigma_2^2 + 2F_{26}\sigma_2\sigma_6 + F_{66}\sigma_6^2 = 1
\end{aligned}
\tag{2}
$$

This expression can be further simplified by assuming that the failure criterion is not influenced by the sign of the shear stress. Then components of f containing linear terms σ_6 must vanish (i.e., $F_6 = F_{16} = F_{26} = 0$) and Eq. 2 reduces to:

$$
f = F_1\sigma_1 + F_2\sigma_2 + F_{11}\sigma_1^2 + 2F_{12}\sigma_1\sigma_2 + F_{22}\sigma_2^2 + F_{66}\sigma_6^2 = 1 \tag{3}
$$

The interaction term is constrained by a stability condition that ensures the failure surface will be ellipsoidal—not an open-ended hyperboloid—and intercept each stress axis [Tsai and Wu, 1971]:

$$
F_{11}F_{22} - F_{12}^2 \geq 0 \tag{4}
$$

The tensor components F_1, F_{11} can be related to the normal stress allowables σ_1^C, σ_1^T by assuming a uniaxial stress field that causes failure. Let $\sigma_1 = \sigma_1^T$, $\sigma_2 = \sigma_6 = 0$; then Eq. 3 becomes:

$$
f = F_1\sigma_1^T + F_{11}\left(\sigma_1^T\right)^2 \tag{5}
$$

Similarly, for $\sigma_1 = -\sigma_1^C$, $\sigma_2 = \sigma_6 = 0$, Eq. 3 becomes:

$$
f = -F_1\sigma_1^C + F_{11}\left(\sigma_1^C\right)^2 \tag{6}
$$

Solving Eqs. 6 and 5 yield expressions for the tensor components F_1, F_{11}:

$$
\begin{aligned}
F_1 &= \frac{1}{\sigma_1^T} - \frac{1}{\sigma_1^C} \\
F_{11} &= \frac{1}{\sigma_1^T\sigma_1^C}
\end{aligned}
\tag{7}
$$

The tensor components F_2, F_{22} can be derived in a similar manner and related to the transverse stress allowables σ_2^C, σ_2^T:

$$
F_2 = \frac{1}{\sigma_2^T} - \frac{1}{\sigma_2^C}
$$

$$
F_{22} = \frac{1}{\sigma_2^T \sigma_2^C} \tag{8}
$$

and F_{66} can be related to shear stress allowable σ_6^F:

$$
F_{66} = \left(\frac{1}{\sigma_6^F} \right)^2 \tag{9}
$$

where we assume that the shear stress failure is independent of the sign of the shear stress, i.e., $\sigma_6^C = \sigma_6^T = \sigma_6^F$.

The experimental determination of the normal stress interaction term F_{12} presents some difficulty since it involves testing a specimen in a combined stress state with σ_1, σ_2 nonzero, and any number of combined stresses is possible. See [Pipes and Cole, 1973], [Tsai and Wu, 1975], and [Narayanaswami and Adelman, 1977] for relevant discussion of this problem.

Another approach used is to choose F_{12} so that the Tsai-Wu theory degenerates to the von Mises-Hencky theory for isotropic plane stress:

$$
\left(\frac{\sigma_1}{\sigma^F} \right)^2 - \frac{\sigma_1 \sigma_2}{\left(\sigma^F \right)^2} + \left(\frac{\sigma_2}{\sigma^F} \right)^2 = 1 \tag{10}
$$

where σ^F is the isotropic yield stress and σ_1, σ_2 are principal stresses. To transform the Tsai-Wu criterion to this form, assume the stress allowables are equal to the yield stress, $\sigma_1^T = \sigma_1^C = \sigma_2^T = \sigma_2^C = \sigma^F$; then the tensor components F_1, F_2 are zero. Also, assume that the stress state corresponds to the principal stresses, so σ_6 vanishes. Then Eq. 3 becomes:

$$
f = F_{11}\sigma_1^2 + 2F_{12}\sigma_1\sigma_2 + F_{22}\sigma_2^2 = 1 \tag{11}
$$

Comparing Eqs. 10 and 11 and recalling the stability criterion for F_{12} given by Eq. 4 suggests the following form for F_{12}:

$$
F_{12} = -\tfrac{1}{2}\sqrt{F_{11}F_{22}} \tag{12}
$$

This last expression was used to compute F_{12} in the present study.

2.2 Composites Lab

A laminated composite is constructed from thin orthotropic plies that can be rotated around their normal axis, as illustrated in Fig. 7. The Composites Lab has a laminate definition area to build the laminate, including the material name (picked from materials defined in the Materials Lab), thickness, and orientation of each ply. The laminate stress analysis is completed using classical lamination theory (see [Jones 1975]), which predicts the stiffness of the laminate and enables stresses to be recovered in individual plies. The laminate stiffnesses are used to predict the equivalent engineering properties for the laminate, which are used to compute the modulus weighted section properties in the Beam Lab. The methodology for computing these equivalent properties is described in Section 2.2.1.

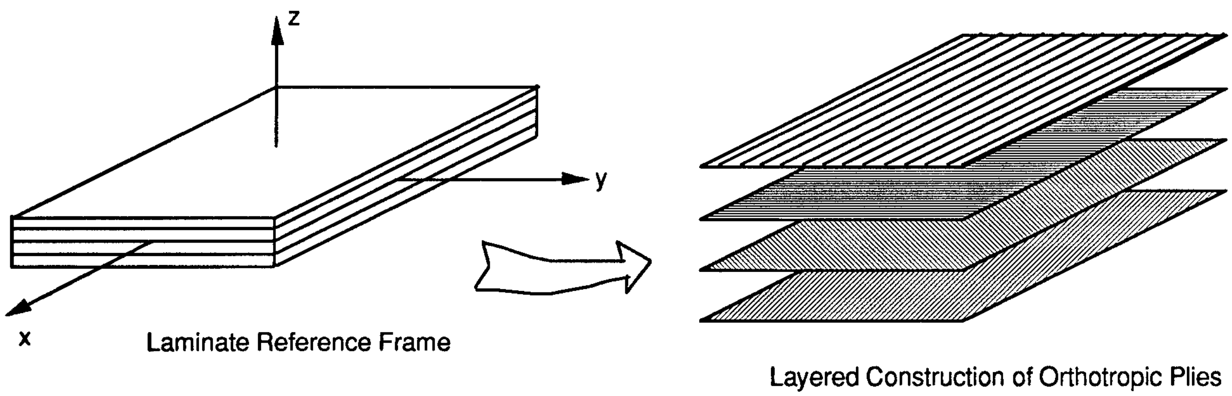


Figure 7—Laminate Constructed from Thin, Arbitrarily-Oriented, Orthotropic Plies

Another function of the Composites Lab is to compute the allowable load for the laminate. This is done by identifying the critical ply which fails first for a given loading configuration, the first-ply failure (FPF) location. The ply-level stresses are used to evaluate the failure criterion at several locations in each ply to identify the FPF location, as described in Section 2.2.2.

Finally, the Composites Lab will plot the through-thickness (ply-level) stresses and the Tsai-Wu failure surface at the computed allowable load. The information is displayed graphically, as shown in Fig. 5, and includes three plots for σ_1, σ_2, σ_6. A slider allows you to specify a location in the laminate z-direction, and the stress values at that location are printed. Additionally, the failure surface—a function of the stress state—is recomputed whenever the slider is moved.

2.2.1 Effective Laminate Engineering Properties

The laminate constitutive relationships can be derived via CLT as:

$$\begin{Bmatrix} N \\ M \end{Bmatrix} = \begin{bmatrix} A & B \\ B & D \end{bmatrix} \begin{Bmatrix} \varepsilon^{\circ} \\ \kappa \end{Bmatrix} \tag{13}$$

where {N}, {M} are the laminate stress resultants and couples; [A], [B], [D] are the extensional, extension-bending coupling, and bending stiffness matrices; and ε°, κ are the midplane strains and curvature. We assume that the laminate is symmetric, for which the components of the [B] matrix are all zero. Then considering just extensional effects, the laminate stiffness relationships become:

$$\{N\} = [A]\{\varepsilon^{\circ}\} \tag{14}$$

Equation 14 can be inverted to obtain the laminate compliance relationships:

$$\{\varepsilon^{\circ}\} = [A]^{-1}\{N\} \tag{15}$$

The stress resultants, {N}, can be converted to equivalent stresses acting on the laminate by normalizing by the laminate thickness, t

$$\{\bar{\sigma}\} = \frac{1}{t}\{N\} \tag{16}$$

Substituting Eq. 16 into Eq. 15 yields:

$$\{\varepsilon^{\circ}\} = \left[A^{*}\right]\{\bar{\sigma}\} \tag{17}$$

where

$$\left[A^{*}\right] = t[A]^{-1} \tag{18}$$

The equivalent engineering properties may be obtained from Eq. 17 by equating the components of $[A^{*}]$ to the usual plane stress compliance matrix

$$[S] = \begin{bmatrix} \frac{1}{E_x^{*}} & -\frac{v_{yx}^{*}}{E_y^{*}} & 0 \\ -\frac{v_{xy}^{*}}{E_x^{*}} & \frac{1}{E_y^{*}} & 0 \\ 0 & 0 & \frac{1}{G_{xy}^{*}} \end{bmatrix} \tag{19}$$

where E_x^{*}, E_y^{*} are the equivalent elastic moduli, G_{xy}^{*} is the equivalent shear modulus, and v_{xy}^{*} is the equivalent Poisson's ratio. Then equating $[A^{*}]$ and [S] yields

$$
\begin{aligned}
E_x^* &= \frac{1}{A_{11}^*} \\
E_y^* &= \frac{1}{A_{22}^*} \\
v_{xy}^* &= -\frac{A_{12}^*}{A_{11}^*} \\
G_{xy}^* &= \frac{1}{A_{33}^*}
\end{aligned}
\tag{20}
$$

2.2.2 Allowable Laminate Load Calculation

The allowable load is computed for a particular component (e.g. N_x, N_y, N_{xy}, M_x, M_y, M_{xy}) in two steps. First, the critical FPF location is found for a unit load in the specified direction. Then the unit load is scaled such that the failure criterion is exactly satisfied at the critical location.

The critical FPF location is determined by recovering the ply-level stresses for a unit load, $\hat{\sigma}_I$, and evaluating the failure criterion

$$f = f\left(F_I, F_{IJ}, \hat{\sigma}_I\right) \tag{21}$$

The allowable load is computed by scaling the unit load such that the FPF criterion is exactly satisfied, i.e.

$$f = f\left(F_I, F_{IJ}, \lambda\hat{\sigma}_I\right) = 1 \tag{22}$$

The Tsai-Wu failure criterion can be written in terms of the scaled stresses, $\lambda\hat{\sigma}_I$, as:

$$F_1(\lambda\hat{\sigma}_1) + F_2(\lambda\hat{\sigma}_2) + F_{11}\lambda^2\hat{\sigma}_1^2 + F_{22}\lambda^2\hat{\sigma}_2^2 + F_{12}\lambda^2\hat{\sigma}_1\hat{\sigma}_2 + F_{66}\lambda^2\hat{\sigma}_6^2 = 1 \tag{23}$$

Rearranging Eq. 23 in the form of a quadratic polynomial in terms of the scale factor λ yields:

$$a\lambda^2 + b\lambda + c = 0 \tag{24}$$

where

$$
\begin{aligned}
a &= F_{11}\hat{\sigma}_1^2 + F_{22}\hat{\sigma}_2^2 + F_{12}\hat{\sigma}_1\hat{\sigma}_2 + F_{66}\hat{\sigma}_6^2 \\
b &= F_1\hat{\sigma}_1 + F_2\hat{\sigma}_2 \\
c &= -1
\end{aligned}
$$

The roots of Eq. 24 yield two possible projections of $\hat{\sigma}$ onto the failure surface. Currently, V-LAB reports the maximum of the two roots. Several points within each ply are tested, and the critical location is determined by the *lowest* value of λ, i.e. the lowest load that causes FPF.

2.3 Beam Lab

The Beam Lab provides capabilities for defining and analyzing composite beam sections, including rectangular, box, I-section, C-channel, and cylindrical beam sections (see Figure 8). Beams are defined by specifying geometry and material data for each "branch" of a beam section, e.g. the top flange, web, and bottom flange of an I-beam. The beam materials are designated by assigning a composite laminate picked from those defined in the Composites Lab. The lab includes functions to compute the modulus-weighted section properties and allowable loads, described in Sections 2.3.2 and 2.3.3.

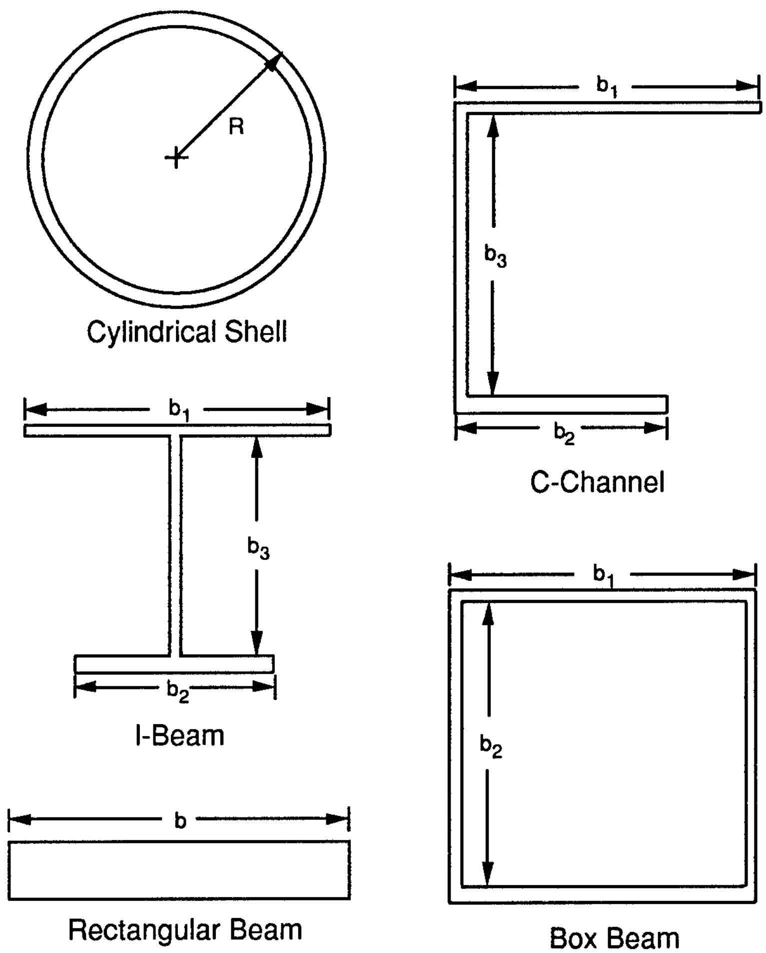


Figure 8—V-LAB Supports a Variety of Thin-Walled Beam Cross Sections

2.3.1 *Review of Beam Theory*

The beam displacement field is defined by a single translational degree of freedom ($\overline{u}$) and two rotational DOFs (θ_x, θ_y):

$$u = \overline{u} - y\theta_z + z\theta_y \tag{25}$$

The extensional strain can be written as

$$\varepsilon_x = \overline{\varepsilon}_x + y\kappa_z + z\kappa_y \tag{26}$$

where

$$\begin{aligned} \overline{\varepsilon}_x &= \partial\overline{u}/\partial x \\ \kappa_y &= \partial\theta_y/\partial y \\ \kappa_z &= -\partial\theta_z/\partial z \end{aligned} \tag{27}$$

The stress-strain relationship can be written as follows, assuming the material is linearly elastic and isotropic

$$\sigma_x = E\varepsilon_x = E\left(\varepsilon_x + y\kappa_z + z\kappa_y\right) \tag{28}$$

where E is the elastic modulus. Define the following beam stress resultants, which are energetically conjugate to the beam strain measures given by Eq. 27:

$$\begin{Bmatrix} N_x \\ M_z \\ M_y \end{Bmatrix} = \int_A \sigma_x \begin{Bmatrix} 1 \\ y \\ z \end{Bmatrix} da \tag{29}$$

Substituting Eq. 28 in 29 yields

$$\begin{Bmatrix} N_x \\ M_z \\ M_y \end{Bmatrix} = \begin{bmatrix} EA & EAy_c & EAz_c \\ & EI_{yy} & EI_{yz} \\ \text{sym.} & & EI_{zz} \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \kappa_z \\ \kappa_y \end{Bmatrix} \tag{30}$$

where

$$\left(EA, EAy_c, EAz_c, EI_{yy}, EI_{yz}, EI_{zz}\right) = \int_A E\left(1, y, z, y^2, yz, z^2\right) da \tag{31}$$

2.3.2 Beam Section Calculations

The beam cross-section is comprised of an arbitrary number of laminate sections (e.g. 4 sections in a box beam, 3 sections in an I-beam, see Figure 8). The beam section properties defined in Eq. 31 were approximated using effective elastic moduli, E^*, for each laminate in the beam cross-section, as computed by Eq. 20. The integral expression in Eq. 31 was approximated as:

$$\int_A E(1, y, z, \ldots)da = \sum_{i=1}^{N} E_i^*(1, y_i, z_i, \ldots)A_i \tag{32}$$

where N is the number of sections in the beam; E_i^* is the effective modulus for the laminate in section i; y_i, z_i are the coordinates of the section centroid; and A_i is the section area. The inertia properties (I_{yy}, I_{yz}, I_{zz}) are computed with respect to a local beam reference frame, and then transformed to the centroidal axes defined by

$$\begin{aligned} y_c &= {EAy_c}/{EA} \\ z_c &= {EAz_c}/{EA} \end{aligned} \tag{33}$$

2.3.3 Beam Failure Analysis

The critical beam failure load is determined by testing each laminate in the beam cross-section. Each ply within the laminate is tested using a FPF failure criterion, as described for the Composites Analysis Lab. The critical FPF value is computed for a unit load in the direction specified by the user (e.g. N_x, M_x, M_y). Then the load is scaled so that the FPF value exactly equals one—indicating the onset of failure—and the scaled load is reported to the user. The scaling is exactly the same as described for the Composites Lab, see Section 2.2.2.

The beam strain resultants are found by inverting Eq. 30:

$$\begin{Bmatrix} \varepsilon_x \\ \kappa_z \\ \kappa_y \end{Bmatrix} = \begin{bmatrix} EA & EAy_c & EAz_c \\ & EI_{yy} & EI_{yz} \\ \text{sym.} & & EI_{zz} \end{bmatrix}^{-1} \begin{Bmatrix} N_x \\ M_z \\ M_y \end{Bmatrix} \tag{34}$$

Then the pointwise strains are computed by stepping through each ply in the laminate at a particular point in the beam. Recall pointwise strains are given by

$$\varepsilon_x = \bar{\varepsilon}_x + z\kappa_y + y\kappa_z \tag{35}$$

The corresponding pointwise stress is found by the reduced constitutive relationships for a thin ply (assuming plane stress):

$$\sigma_x = \overline{Q}_{11}\varepsilon_x \qquad (36)$$

where we assume $\{\varepsilon\}^T=\{\varepsilon_x \ \ 0 \ \ 0\}$ and $\{\sigma\}^T=\{\sigma_x \ \ 0 \ \ 0\}$, i.e. only 1D strains are acting on the beam, and we ignore σ_y, σ_{xy} which have no meaning in the 1D beam theory.

The global stress is transformed to the material axes using the transformation equations for an off-axis ply [Jones 1975]:

$$\{\sigma\}_1 = [T]\{\sigma\}_x \qquad (37)$$

The stresses are used to evaluate the failure criterion and allowable load using the same methodology described in Section 2.2.2. The critical FPF location and allowable load is determined by checking each laminate section at several points in the beam cross-section.

3.0 V-LAB Software System

3.1 Software Framework Overview

An overview of the V-LAB software framework is shown in Figure 9. The dynamic link library VLABWIN.DLL contains the graphical user interface code for the three main analysis windows (Materials Lab, Composites Lab, and Structures Lab) as well as the Micromechanics Lab window. Including the code for all the screens in the same library enables them to seamlessly access each other, which makes it possible to open or "call" any one screen from another and then destroy the original "calling" window. This creates the appearance that there is one main analysis screen that changes appearance in order to perform either materials analysis, composite analysis, or structural analysis depending on the user's selection. This is important because it directs the focus of the program and prevents it from performing overly confusing or conflicting tasks (such as redefining the properties of a material while simultaneously analyzing a beam section made of that material).

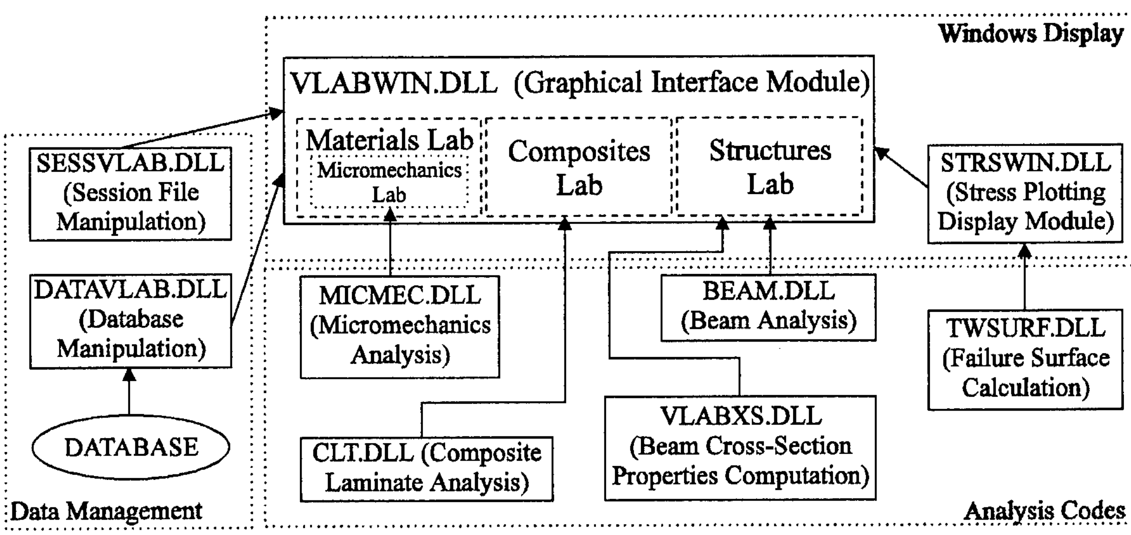


Figure 9—V-LAB Software Framework

The V-LAB database is accessed through the module DATAVLAB.DLL. This dynamic link library contains all the function calls required by the program for data retrieval and storage. The main program module, VLABWIN.DLL, sends function calls to DATAVLAB.DLL to retrieve the data required by all the analysis modules, thus maintaining the independence of all the analysis modules so that they can be used by other programs that do not have access to the V-LAB database. The design and implementation of the database and the data storage/retrieval library is discussed in greater detail in the next section.

In order to preserve the integrity of the V-LAB database, it is never actually modified. Instead, new data is stored in temporary files identified by the current program "session." The temporary files are created and identified by the SESSVLAB.DLL library, which is accessed by the main program module whenever the File menu options ("New," "Open," "Close," etc.) are selected by the user. Together with DATAVLAB.DLL, this pair of libraries provides all the data handling functionality of the program: SESSVLAB.DLL handles file location and structure; and DATAVLAB.DLL manipulates the data in the files.

There are four analysis modules that are accessed by the main program module: MICMEC.DLL, CLT.DLL, VLABXS.DLL, and BEAM.DLL. These were coded in FORTRAN and have been compiled as dynamic link libraries which allows them to be accessed by other programs. The database access provided by V-LAB increases the effectiveness of these modules as the program handles all the database input and output operations for these codes. MICMEC.DLL is accessed by the Micromechanics Lab sub-screen to perform micromechanics analysis of materials; CLT.DLL is accessed by the Composites Lab screen to perform stress analysis of composite laminates; VLABXS.DLL is accessed by the Structures Lab to calculate geometric properties of the cross-section of a structural component; and BEAM.DLL is accessed by the Structures Lab to perform stress analysis of the structural component.

VLABWIN.DLL links to one other windows library, STRSWIN.DLL, which contains the code for graphical display of the failure surfaces and stress distributions. It contains the function "PlotStressState" which is called by all three of the main analysis windows. This in turn links to the library TWSURF.DLL, which calculates the Tsai-Wu failure surface information.

3.2 V-LAB Database Design

The database design process consisted of three steps: (1) identification of data requirements, (2) conceptual schema design, (3) relational database implementation. The first step was performed during the interface design process. As we drew the screen concepts we defined how the software should perform. For example, selecting a specific material type from the "Material Type" combo box should limit the potential selections in the "Material Name" combo box to be materials of the selected type. Thus a relationship was identified between the entities *material* and *material type*: A material must have one and only one material type, though a material type may describe any number of materials.

Once all the required data entities, as well as the relationships between them, were identified, a conceptual schema diagram was developed using the binary (one-to-one) modeling method. A portion of the V-LAB Phase I conceptual schema is shown in Figure 10:

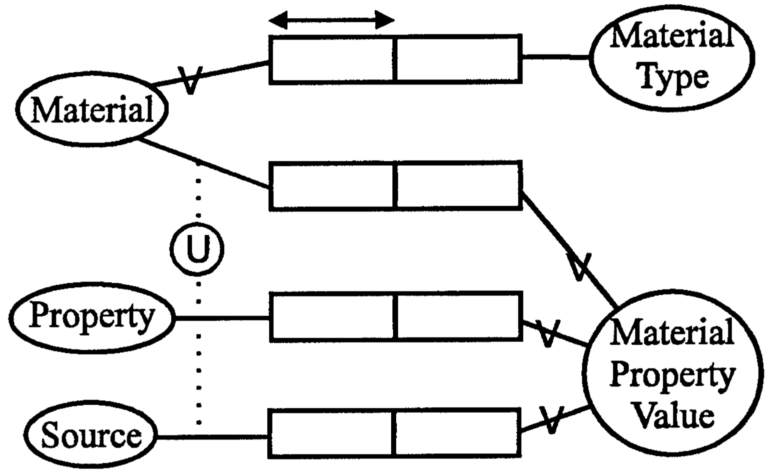


Figure 10—A Portion of the V-LAB Conceptual Database Schema Representing the Relationship Between Materials and their Properties

Data entities are represented as ellipses, and the relationships between them are indicated by the boxes through which they are connected. A *mandatory* role constraint (e.g, "A material *must* have . . . .") is represented by a V symbol over the line connecting the entity to the relationship, and a *uniqueness* constraint (e.g., "A material has *one and only one* material type . . . .") is represented by an arrow over the unique role in the relationship box.

In some cases a uniqueness constraint applies to a combination of roles and not a single role. For example, suppose that a handbook lists the value of E_1 for material AS4/3502 to be 19.60 MSI, while a lab test results in a value of 18.5 MSI and a micromechanics analysis code predicts a value of 20.2 MSI. (It is often helpful to examine the data in tabular form, as shown below.)

Row #	Material	Property	Value	Source
1	AS4/3502	E1	19.6	MIL HDBK 17
2	AS4/3502	E1	18.5	Test #435
3	AS4/3502	E1	20.2	MICMEC.EXE
. . .	. . .	. . .	. . .	. . .

Although each row lists a different value for the same property of the same material, each row is still valid data because it simply states that a specific source provided each value. We could also expect to see the following rows of data added to the table:

Row #	Material	Property	Value	Source
1	AS4/3502	E1	19.6	MIL HDBK 17
2	AS4/3502	E1	18.5	Test #435
3	AS4/3502	E1	20.2	MICMEC.EXE
4	AS4/3502	E2	1.45	MIL HDBK 17
5	AS4/3501-6	E1	21.8	MICMEC.EXE
6	AS4/3502	E1	19.6	Test #502
. . .	. . .	. . .	. . .	. . .

Row #4 duplicates the material name and source name from Row #1 yet it lists a different value. However, there is no conflict between the rows because the property name has changed. Row #5 duplicates the property name and source name from Row #3 yet it lists a different value. However, there is no conflict between the rows because the material name has changed. Row #6 duplicates the material name, property name, and value from Row #1, and at first glance it may appear that this row simply provides duplicate data that adds no real information to the table. However, the name of the source has changed. The row actually tells us that a laboratory test was able to duplicate the value found in the military handbook. Therefore none of these rows conflict with one another or duplicate one another. However, the following additional rows would indicate a data integrity problem:

Row #	Material	Property	Value	Source
1	AS4/3502	E1	19.6	MIL HDBK 17
2	AS4/3502	E1	18.5	Test #435
3	AS4/3502	E1	20.2	MICMEC.EXE
4	AS4/3502	E2	1.45	MIL HDBK 17
5	AS4/3501-6	E1	21.8	MICMEC.EXE
6	AS4/3502	E1	19.6	Test #502
7	AS4/3502	E2	2.10	MIL HDBK 17
8	AS4/3502	E1	20.8	MICMEC.EXE
9	AS4/3502	E1	18.5	Test #435
. . .	. . .	. . .	. . .	. . .

Row #7 has the same material name, property name, and source name as Row #4, although the value is different. This indicates either that one of the rows was inserted in error, or that the military handbook provides two different values for one specific property of one specific material. Likewise, Row #8 has the same material name, property name, and source name as Row #3, although the value is different. This indicates either that one of the rows was inserted in error, or that the micromechanics analysis code resulted in two different results for the same input. Row #9 has the same material name, property name, source name, and value as Row #2. While this does not represent conflict, it is unnecessary and there is no benefit to having it in the table.

Thus a material property value is uniquely identified by the combination of material, property, and source. A single source may provide one and only one value for a specific property of a specific material (or for multiple materials). Multiple sources may provide the same value for the same property of the same material. This combined uniqueness constraint is represented in the conceptual schema diagram by the dotted line with the encircled U symbol connecting the three

roles. The combination of these roles is the unique key to the relationship; all other roles represent nonkey attributes.

Because the conceptual schema diagram represents established relationships among the data required by V-LAB, new relationships representing new information requirements can be added to the diagram without modifying the existing relationships. This is an important feature of the V-LAB database design that accommodates extending V-LAB to add new analysis codes. For example, we wish to require that the constituent matrix and fiber be identified for each material stored in the database. The fiber and matrix materials will have property values that differ from those of the composite (fiber/matrix) material. Also, the material definition is incomplete without the fiber radius r and the diamond RVE parameters d_1 and d_2. These additions are shown in Figure 11:

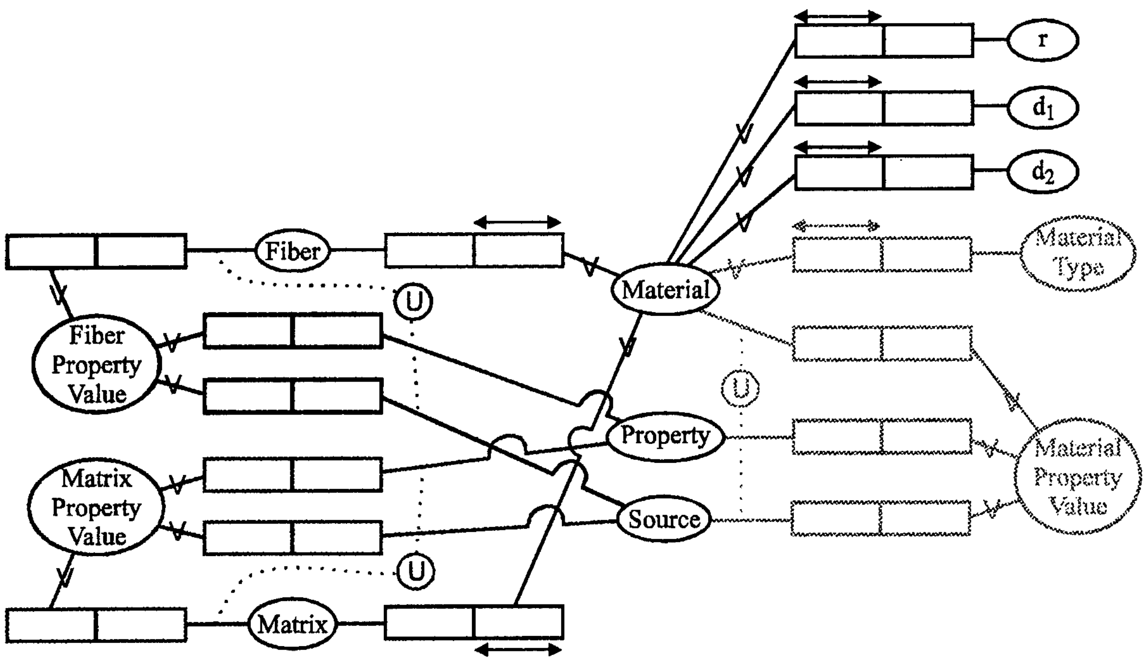


Figure 11—Expansion of the Conceptual Database Schema to Represent Additional Relationships

Once all information requirements are identified and a complete conceptual schema diagram is developed, the diagram is converted to an optimized relational database schema using the Optimal Normal Form algorithm [Nijssen and Haplin, 1989]. A table is said to be in a particular normal form if it satisfies a certain set of constraints related to the data integrity issues discussed above. Generally the process of normalizing relationships is an attempt to have all nonkey attributes be both mutually independent on one another and fully dependent on the primary key.

The partial conceptual schema shown above is converted into the following tables, with primary unique keys surrounded by bold rectangles:

Material	Material Type	Fiber	Matrix	r	d1	d2

Material	**Property**	**Source**	Value

Fiber	**Property**	**Source**	Value

Matrix	**Property**	**Source**	Value

The database design process can be summarized as follows: During the software design process, the information requirements of the program (i.e., what information it needs) are identified. During the conceptual schema diagram development, relationships between the data items are identified *independent* of the software design. Each relationship is analyzed to ensure that the resulting database tables represent basic information that is required to define materials, laminates, and beam structures. The software developer must then structure queries in order to retrieve information from the database that meets these requirements. This is an inversion of the likely natural process, which would be to form a query based on the information requirements and design the table to reflect the query.

The process used to design the V-LAB database safeguards the data integrity by preventing conflicts, and ensures the database is extensible to accommodate software changes. A single database can be used by a number of different software modules because it has not been designed explicitly for use by any single one. This has an important implication for adding new analysis modules to V-LAB. Additional information requirements for the new modules can be added to the original conceptual schema diagram without invalidating existing relationships. Existing tables will not be broken apart, although new columns may be added to those tables or completely new tables may be added to the database. Therefore all the original queries remain valid, and new ones may be added.

3.3 Database Entities and Utilities

The data structures of the primary database tables used in the Phase I version of V-LAB are listed below, with the table names underlined, and the primary unique key shown in boldface. Descriptions of columns (fields) or tables may follow its listing in brackets.

```
MAT.DBF                                         [Material data]
Field  Field Name   Type         Width  Dec
    1  MATERIAL     Character       32      [material name]
    2  FIBER        Character       32      [fiber name for given material]
    3  MATRIX       Character       32      [matrix name for given material]
    4  MATERTYPE    Character       32      [type of material]
    5  MANUFACTUR   Character       32      [name of manufacturer]
    6  RENGVALUE    Numeric         13    6 [fiber radius in inches]
    7  D1ENGVALUE   Numeric         13    6 [horizontal dimension of rve]
```

Field	Field Name	Type	Width	Dec	
8	D2ENGVALUE	Numeric	13	6	[vertical dimension of rve]
** Total **			200		

MATPRP.DBF [Material Property data]

Field	Field Name	Type	Width	Dec	
1	**MATERIAL**	Character	32		
2	**PROPERTY**	Character	32		[property identifier]
3	**SOURCE**	Character	32		[source name]
4	ENGVALUE	Numeric	13	6	[value in English units]
** Total **			110		

FIBPRP.DBF [Fiber Property data]

Field	Field Name	Type	Width	Dec
1	**FIBER**	Character	32	
2	**PROPERTY**	Character	32	
3	**SOURCE**	Character	32	
4	ENGVALUE	Numeric	13	6
** Total **			110	

MTXPRP.DBF [Matrix Property data]

Field	Field Name	Type	Width	Dec
1	**MATRIX**	Character	32	
2	**PROPERTY**	Character	32	
3	**SOURCE**	Character	32	
4	ENGVALUE	Numeric	13	6
** Total **			110	

PROP.DBF [Property data]

Field	Field Name	Type	Width	Dec	
1	**PROPERTY**	Character	32		
2	UNITS	Character	32		[units ID for given property]
3	BITMAP	Character	12		[graphics filename for property]
4	DESCRIPT	Character	32		[description of property]
** Total **			109		

UNIT.DBF [Unit data]

Field	Field Name	Type	Width	Dec	
1	**UNITS**	Character	32		
2	ENGDISP	Character	32		[English display text]
3	METDISP	Character	32		[metric display text]
4	ENGTOMET	Numeric	13	6	[English-to-metric conversion]
** Total **			110		

LAM.DBF [Laminate data]

Field	Field Name	Type	Width	Dec	
1	**LAMINATE**	Character	32		[laminate name]
2	**LAYER**	Numeric	2		[layer number]
3	MATERIAL	Character	32		[material used for given layer]
4	ENGORIENT	Numeric	13	6	[orientation in degrees]
5	ENGTHICK	Numeric	13	6	[thickness in inches]
** Total **			93		

LAMSEQ.DBF [Laminate Sequence data]

Field	Field Name	Type	Width	Dec	
1	**LAMINATE**	Character	32		
2	STACKSEQ	Character	32		[stacking sequence]
** Total **			65		

SECT.DBF [Section Type data]

Field	Field Name	Type	Width	Dec	
1	**BEAMSECT**	Character	32		[section type name]
2	IMAGEFILE	Character	12		[graphics filename for section]
3	LOCATEFILE	Character	12		[second graphics filename]

Field	Field Name	Type	Width	Dec	
4	XPIXELS	Numeric	3		[horizontal size of graphic]
5	YPIXELS	Numeric	3		[vertical size of graphic]
** Total **			63		

<u>SECTPM.DBF</u> [Section Parameter data]

Field	Field Name	Type	Width	Dec	
1	**BEAMSECT**	Character	32		
2	**SUBSECTION**	Character	32		[name of subsection for type]
3	PARAMETER	Character	32		[parameter name for subsect. length]
4	PIXELS	Numeric	3		[subsection length in graphic]
5	THICKPIX	Numeric	3		[subsection thickness in graphic]
** Total **			103		

<u>SECTLOC.DBF</u> [Section Location data]

Field	Field Name	Type	Width	Dec	
1	**BEAMSECT**	Character	32		
2	**LOCATION**	Character	1		[name of predef. locat. on section]
3	XPIXELS	Numeric	4		[horizontal location on graphic]
4	YPIXELS	Numeric	4		[vertical location on graphic]
** Total **			42		

<u>BEAM.DBF</u> [Beam data]

Field	Field Name	Type	Width	Dec	
1	**BEAM**	Character	32		[name of beam]
2	BEAMSECT	Character	32		[section type of given beam]
** Total **			65		

<u>BEAMPRP.DBF</u> [Beam Property data]

Field	Field Name	Type	Width	Dec	
1	**BEAM**	Character	32		
2	**SUBSECTION**	Character	32		
3	LAMINATE	Character	32		[laminate used for given subsection]
4	PARENGVAL	Numeric	13	6	[English value of subsect. param.]
** Total **			110		

For purposes of data retrieval and manipulation, a dynamic-link library (DLL) was developed with a number of generic database functions. The DLL links to the CodeBase 5.0 library, which provides database management functionality. These generic functions can be rewritten if it is desired to use some other database management library in the future, without affecting the V-LAB program calls to the functions. Some of the functions are listed below in C-language prototype form:

```
int GetTextItemFromFileWhereCols(LPSTR lpszFileName,LPSTR lpszItemName,
    LPSTR lpszItemValue,LPSTR lpszColumnName1,LPSTR lpszColValue1,
    LPSTR lpszColumnName2,LPSTR lpszColValue2,. . .);

int GetIntItemFromFileWhereCols(LPSTR lpszFileName,LPSTR lpszItemName,
    LPSTR lpszColumnName1,LPSTR lpszColValue1,LPSTR lpszColumnName2,
    LPSTR lpszColValue2,. . .);

double GetDoubleItemFromFileWhereCols(LPSTR lpszFileName,
    LPSTR lpszItemName,LPSTR lpszColumnName1,LPSTR lpszColValue1,
    LPSTR lpszColumnName2,LPSTR lpszColValue2,. . .);

int GetTextFromFileWhereTextAndInt(LPSTR lpszFileName,
    LPSTR lpszItemName,LPSTR lpszItemValue,LPSTR lpszColumnName1,
    LPSTR lpszColValue1,LPSTR lpszColumnName2,int iColValue2);

double GetDblFromFileWhereTextAndInt(LPSTR lpszFileName,
```

```
    LPSTR lpszItemName,LPSTR lpszColumnName1,LPSTR lpszColValue1,
    LPSTR lpszColumnName2,int iColValue2);

double SumDoubleItemsFromFileWhereCol(LPSTR lpszFileName,
    LPSTR lpszItemName,LPSTR lpszColumnName,LPSTR lpszColValue);

int CountRowsInFileWhereCol(LPSTR lpszFileName,LPSTR lpszColumnName,
    LPSTR lpszColValue);
```

The first three functions are used to retrieve text, integer, or real data from a nonkey attribute ("lpszItemName") of a table ("File") given the names ("lpszColumnNameX") and values ("lpszColValueX") of the unique key. The unique key can span multiple columns, but all columns are required to contain text data. The next two functions are used if the key consists of two columns: one containing text data and the other containing integer data. The next two functions, "SumDoubleItemsFromFileWhereCol" and "CountRowsInFileWhereCol" perform the mathematical functions implied by their names. These functions are used to perform the data retrieval operations required by the program.

Because this version of V-LAB is developed for Windows, data that is retrieved generally needs to be displayed graphically. For this reason the following functions were developed. These functions combine the CodeBase database management capabilities with calls to Windows custom controls.

```
int FillComboWithItemsFromFile(HWND hDlg,int iComboID,
    LPSTR lpszFileName, LPSTR lpszItemName,LPSTR lpszExtraItemName1,
    LPSTR lpszExtraItemName2,. . .);

int FillCmbWithItemsFromFileWhereCol(HWND hDlg,int iComboID,
    LPSTR lpszFileName,LPSTR lpszItemName,LPSTR lpszColumnName,
    LPSTR lpszColValue);

int CALLBACK FillTblWithItemsFromFileWhereCol(HWND hDlg,int iTableID,
    int iCol,LPSTR lpszFileName,LPSTR lpszColumnName,
    LPSTR lpszColValue,LPSTR lpszItemName1,LPSTR lpszItemName2,. . .);
```

The first function fills a combo-box (in this case, a standard Windows drop-down listbox) with all the text items under a specific column ("lpszItemName") in a table ("lpszFileName"). The function also allows you to retrieve other information from other columns ("lpszExtraItemNameX") on each row. (In this case, pointers to the strings are stored as extra hidden data with each item of the drop-down listbox.) The second function is used for filtering processes. It fills a combo-box with text items from a specific column only from those rows where another column ("lpszColumnName") contains a specific string ("lpszColValue"). This allows the program to perform such operations as limiting a list of material names to include only those of a specific type. The third function is used to fill database information into a tabular spreadsheet control. For this application, the ProtoView DataTable control is used. The function can be rewritten if it is desired to use some other spreadsheet control in the future, without

affecting the V-LAB program calls to the function. The function allows the program to fill consecutive columns of the spreadsheet (beginning with column number "iCol") with values from multiple columns ("lpszItemNameX") from a table where a specific column contains a specific string. The data that is retrieved does not have any specific type, though each column of the spreadsheet control should be configured for the data type that it will receive.

3.4 Session Files

For the purposes of the Phase I software package, we provide the ability for the user to add limited data to the database and save his additions as V-LAB "sessions." However, in order to preserve the integrity of the system data, the user-provided session data is not actually stored in the same tables as the system data. Instead, we identified a limited number of tables that contain data items that we allow the user to provide, and we created a table "template" for each that shares the same data structure but contains no data. When the user creates a session, a copy of each template file is made and stored in the system temporary directory. The V-LAB session (*.VLB) file contains the names of the temporary files and indicates which template was copied. The user-provided data is stored in these temporary files. The database interface code treats these temporary files for the current session as extensions of the system database, and they are bound by the same constraints. For example, the user is not allowed to redefine a material that is defined in the system database, although they are allowed to redefine a material that was defined earlier in the same session. Therefore, if a template is available for the file, it is as if the user can append rows of data to the file (within the mandatory-role constraints and unique-role constraints discussed earlier) and save the additions. To work with just the system data, the user can start a new session and previous additions will be ignored.

Session templates were created for the following tables: MAT.DBF, MATPRP.DBF, FIBPRP.DBF, MTXPRP.DBF, LAM.DBF, LAMSEQ.DBF, BEAM.DBF, and BEAMPRP.DBF. These allow the user to define and save new material definitions, composite laminate definitions, and beam definitions. However, the definitions are limited to the properties and section types that are available in the system. That is, the user cannot define a material using values for properties not already defined in the system, such as "color," or define a beam that has an octagonal section. However, the ability to add data to these eight tables does provide the user a great deal of flexibility. The user can define new materials that are similar to the system materials but with some parameter values modified (e.g. to model defective materials). The user can then define and analyze composite laminates that are composed of both the system materials and the newly-defined materials, and create beams that are composed of these laminates.

The whole process of adding data to the database is simplified through use of the software package. The user never has to manipulate the database using a separate database management system. New materials, laminates, and beams can be defined using Windows edit boxes, listboxes, and spreadsheet controls on the V-LAB screens, and saved using "Save" from the "File" menu. The following functions were added to the database management DLL in order to provide this functionality:

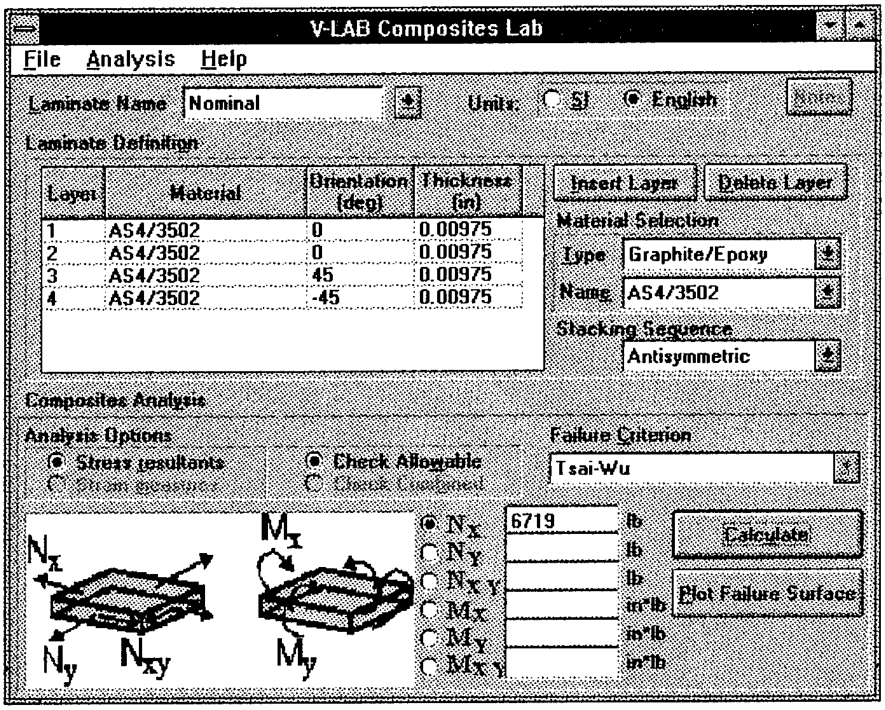
```
int AddRowToMat(LPSTR lpszMaterial,LPSTR lpszFiber, LPSTR lpszMatrix,
    LPSTR lpszMatType,LPSTR lpszManufacturer,double dbRValue,
    double dbD1Value,double dbD2Value);

int AddRowToPrp(LPSTR lpszItem,LPSTR lpszProperty,LPSTR lpszSource,
    double dbValue,int iItemType);

int AddRowToLam(LPSTR lpszLaminate,int iLayer,LPSTR lpszMaterial,
    double dbOrient,double dbThick);

int AddRowToLamSeq(LPSTR lpszLaminate,LPSTR lpszSequence);

int AddRowToBeam(LPSTR lpszBeam,LPSTR lpszSection);

int AddRowToBeamPrp(LPSTR lpszBeam,LPSTR lpszSubsection,
    LPSTR lpszLaminate,double dbParEngVal);

int DeleteRowsFromFileWhereCol(LPSTR lpszFileName,LPSTR lpszColumnName,
    LPSTR lpszColValue);
```

The first function adds data to the duplicate MAT.DBF table, with each parameter of the function representing a field of the table. The second function is used to add data to either MATPRP.DBF, FIBPRP.DBF, or MTXPRP.DBF, with a flag ("iItemType") identifying which table should receive the data, and so on. All the functions check the unique key against the original table to make sure that the user is not attempting to redefine system data. However, the user is allowed to redefine or overwrite his own data. The function "DeleteRowsFromFileWhereCol" deletes all the rows from a duplicate table where a specific column ("lpszColumnName") has a specific value ("lpszColValue") so that that item can be redefined. This function will never delete rows from an original table, because system data is permanent.

4.0 Case Study: Petral Tail Boom

The case study presented here is based on a recent failure investigation of a carbon/epoxy cylindrical shell tail boom performed by the NTSB and the WPAFB Materials Directorate. The post mortem failure investigation of the tail boom revealed that the component had failed as a result of flexural overload. Analysis of the component and subsequent comparison to the drawings revealed only one major material/design discrepancy. The design drawings indicated that the tail boom was to be manufactured using a $[0_2/\pm45/\pm45/0_2]$ layup with a total thickness specified to be 0.078 inches. However, cross-sectional and ply separation analysis showed that the actual layup of the tail boom was just $[\pm45/\pm45]$, with no 0 degree plies, and a total thickness of just 0.043 inches. The problem was analyzed using the Composites and Beam Labs in V-LAB. Two laminates were defined in the Composites Lab (see Figure 12): a "nominal" laminate with a $[0_2/\pm45/\pm45/0_2]$ stacking sequence; and an "as-manufactured" laminate with a $[\pm45/\pm45]$ stacking sequence. The carbon/epoxy material properties were approximated using AS4/3502 data. Note that the Material Lab could be used to define new material properties if they were available.

The Beam Lab was used to model the beam cross-section. The cylindrical shell cross-section type was chosen from the list of available types, and the radius was set to 3.5 inches (the maximum value for the tail boom). The allowable loads were checked for each of the two laminates defined in the Composites Lab (see Figure 13). First the "nominal" laminate was picked for the beam cross-section and the allowable load was calculated by picking the "M_x" component (for flexural loading) and picking the "calculate" button. The allowable load for the nominal laminate was computed as 63,000 lb/in. Next, the "as-manufactured" laminate was picked for the beam, and the allowable load was recalculated as M_x = 9000 lb/in. The allowable bending load decreased by a factor of 7x, indicating a possible cause of failure.



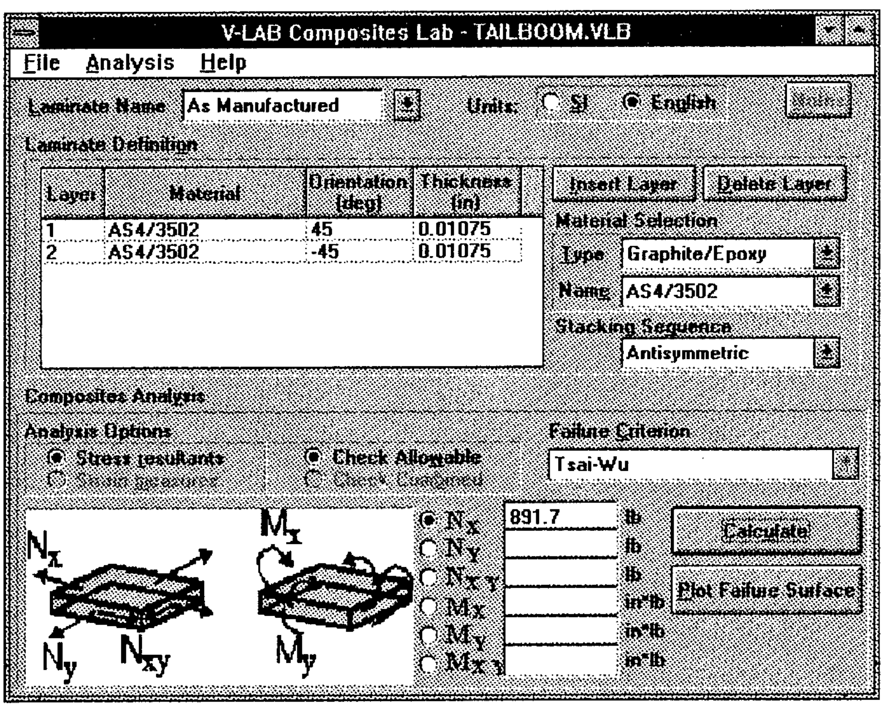
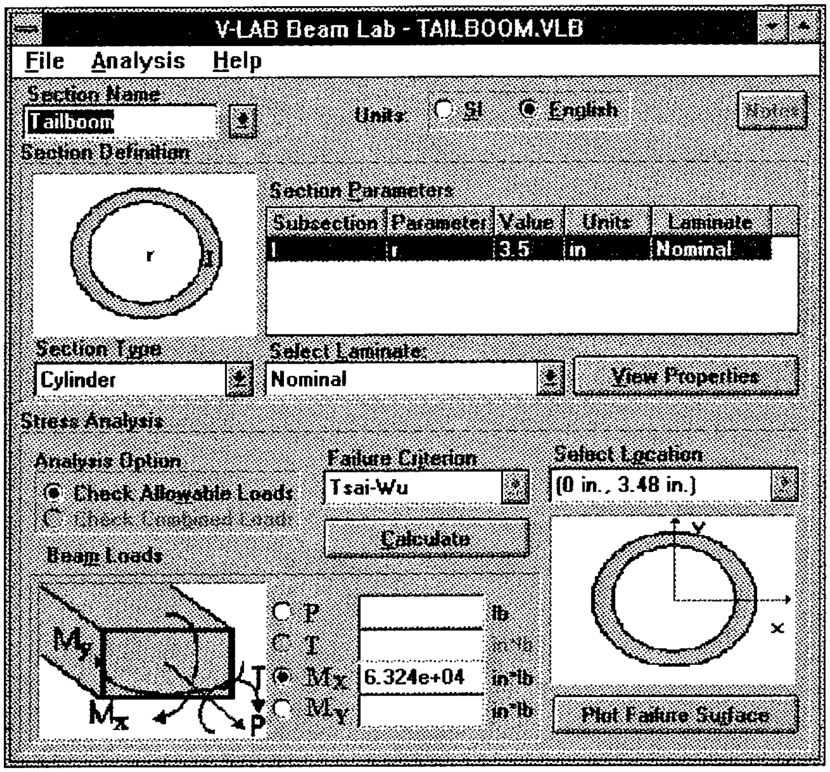


Figure 12—Composites Lab Modeling of Nominal and As-Manufactured Configurations of the Tailboom Laminate



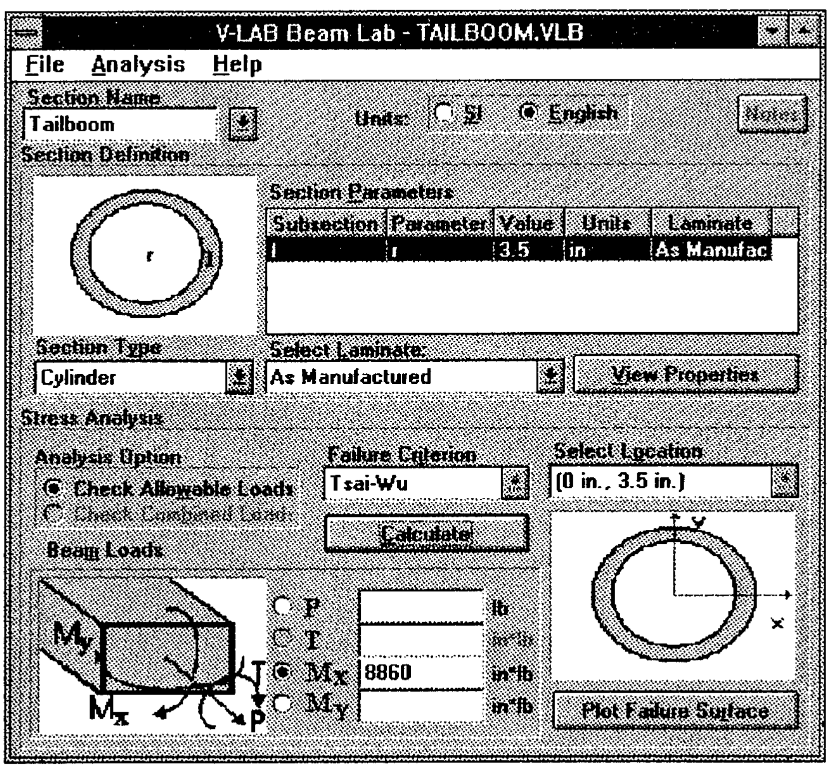


Figure 13—Beam Lab Models the Tailboom Using Cylindrical Shell with Nominal and As-Manufactured Configurations of the Laminate

5.0 References

Averill, R.C. and Carman, G.P. (1991) "Analytical Modeling of Micromechanical Stress Variations in Continuous Fiber-Reinforced Composites," IUTAM Symposium on Local Mechanics Concepts for Composite Material Systems, Blacksburg, Virginia, pp. 27–61.

Jones, M.J. (1975) **Mechanics of Composite Materials**, Hemisphere Publishing Corporation, New York.

Nahas, M.N. (1986) "Survey of Failure and Post-Failure Theories of Laminated Fiber-Reinforced Composites," Journal of Composites Technology and Research, Vol. 8, No. 4, pp. 138–153.

Narayanaswami, R. and Adelman, H.M. (1977) "Evaluation of the Tensor Polynomial and Hoffman Strength Theories for Composite Materials," *Journal of Composite Materials*, Vol. 11, pp. 366–377.

Nijssen, G.M. and Halpin, T.A. (1989) **Conceptual Schema and Relational Database Design: A Fact-Oriented Approach**, Prentice-Hall, Inc., Englewood Cliffs, NJ.

Pipes, R.B. and Cole, B.W. (1973) "On the Off-Axis Strength Test for Anisotropic Materials," *Journal of Composite Materials*, Vol. 7, pp. 246–256.

Tsai, S.W. and Wu, E.M. (1971) "A General Theory of Strength for Anisotropic Materials," *Journal of Composite Materials*, Vol. 5, pp. 58–80.